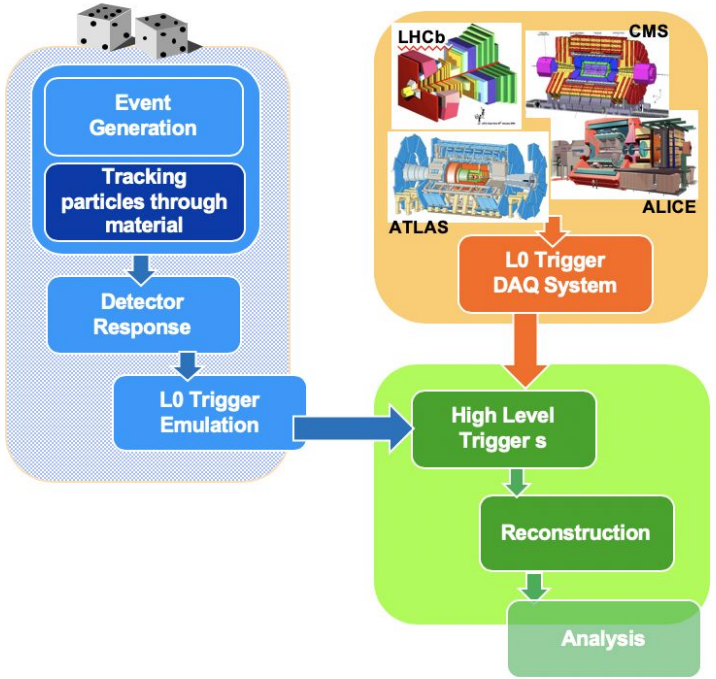
Detector Simulation status and challenges

Gloria Corti[1], Heather Gray[2], Witold Pokorski[1]

[1]CERN, [2]UC Berkeley/LBNL

Introduction

- In High Energy Physics the role of Monte Carlo simulations is to mimic what happens in the experiments
- Monte Carlo data are processed as real data to reconstruct an “image” of the events through measurements by complex detectors comprising many sub-detectors… BUT we know the “truth”!
- Comparing the simulation with what is measured in reality allows to understand the experimental conditions and performance and is a key ingredient in interpreting the results



Why to use Monte Carlo simulations ?

- Aim to simulate events in as much detail as Mother Nature
  - Get average and fluctuations right
  - Make random choices, ~ as in nature
  - An event with n particles involves O(10n) random choices. At LHC: ~ 100 charged and ~ 200 neutral for each collisions (+ intermediate stages) several thousand choices
- This applies also to the particle transport code through the spectrometer and the detectors response
  - “track” the particles in the geometrical setup and have them interact with the matter
  - simulate the detection processes and response of a given detector
  - the interaction events are stochastic and so is the transport process
- A problem well suited for Monte Carlo method simulations
  - computational algorithms relying on repeated random sampling to compute their results

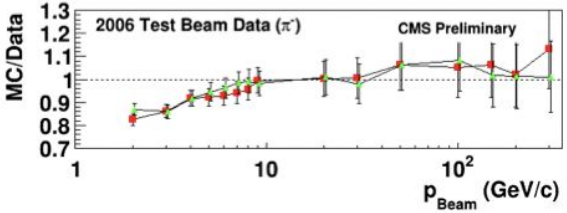
How are MC simulations used ?

- Simulations are present from the beginning of an experiment
  - Simple estimates needed for making detector design choices
  - Develop reconstruction and analysis programs
  - Evaluate physics reach
- They are built up over time
  - Adding/removing details as necessary
- They are used in many different ways
  - Detector performance studies
  - Providing efficiency, purity values for analysis
  - Looking for unexpected effects, backgrounds
  - When theory is non well known compare to various models and accounts for different detector “acceptance”

Examples from CMS

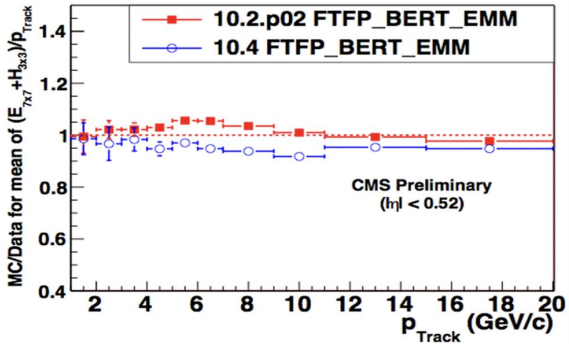
Test Beam 2006

MC-to-data ratio of π energy resolution



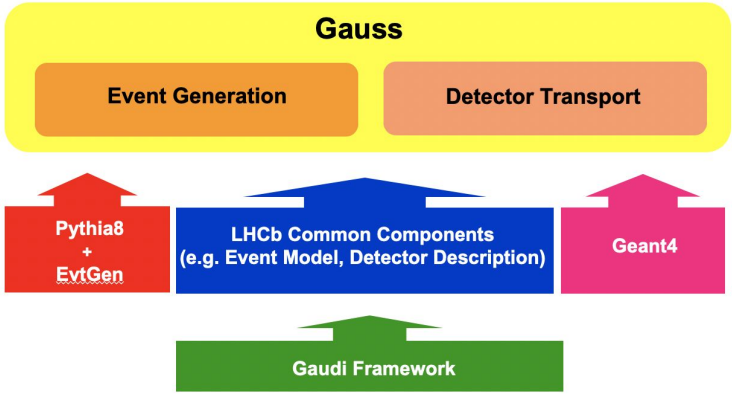
Collider data

MC-to-data ratio of tracker-to-calorimeter single track energy response ratio



Experiments simulation software

- HEP experiments have their own software frameworks
  - Athena (ATLAS), CMSSW (CMS), Gauss on Gaudi (LHCb, MoEDAL, Codex-b), VMC (ALICE, CBM@GSI, Minos), etc.
- and use external packages developed in the physics community for transport in the detectors
  - GEANT4 is the toolkit used by almost all experiments
  - FLUKA mostly used for beam lines and radiation environment



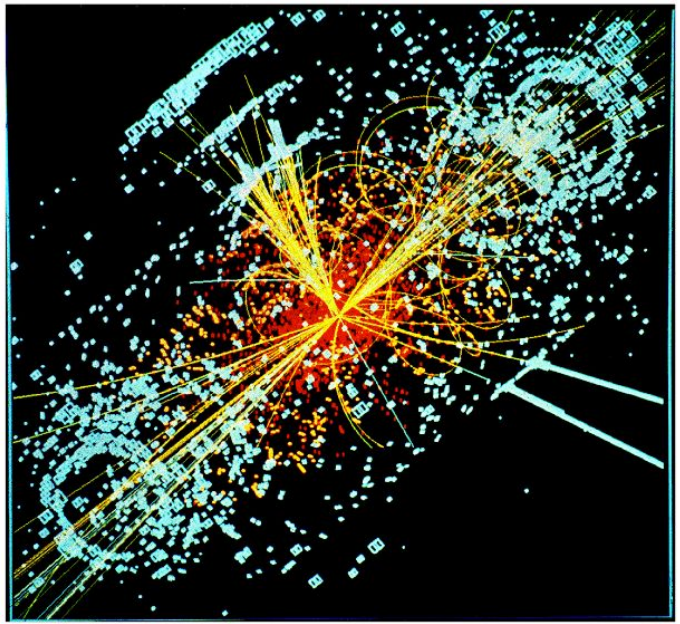
example from LHCb

- Response of the detectors is often in-”house” and requires detectors experts
  - tuned first with test beam data, then with measurements in the experiment

Simulation toolkits - Geant4

- software (C++) toolkit for the Monte Carlo simulation of the passage of particles through matter
  - ‘propagates’ particles through geometrical structures of materials, including magnetic field
  - simulates processes the particles undergo
    - creates secondary particles
    - decays particles
  - calculates the deposited energy along the trajectories and allows to store the information for further processing (‘hits’)

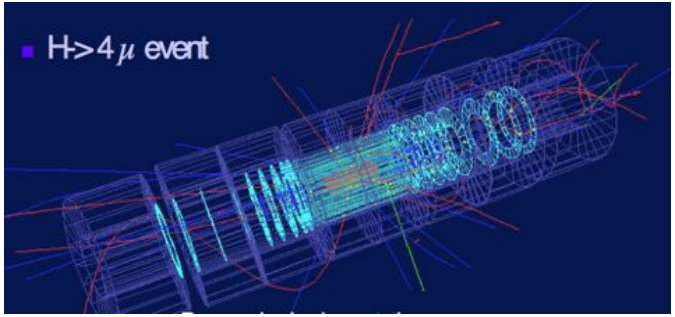
- Other toolkits also exist (FLUKA, MCNP, Penelope, ...) for more specialize (nuclear or electromagnetic applications)



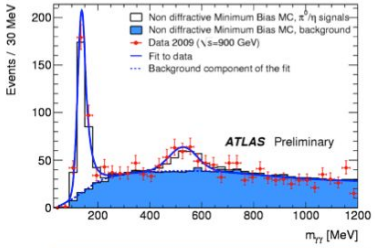
Simulated Higgs event in CMS

Geant4 has been successfully employed for

- Detector design
- Calibration / alignment
- Physics analyses (Higgs discovery!)



GEANT4 Comparisons with the Calorimeters



Invariant mass of pairs of well-isolated electromagnetic clusters.

The π^0 mass is within 0.8 ± 0.6% of expectations.

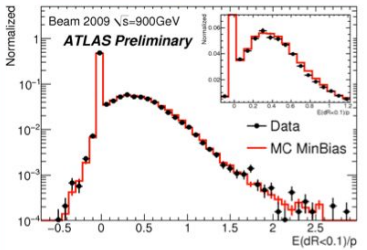
The η^0 mass is within 3 ± 2% of expectations.

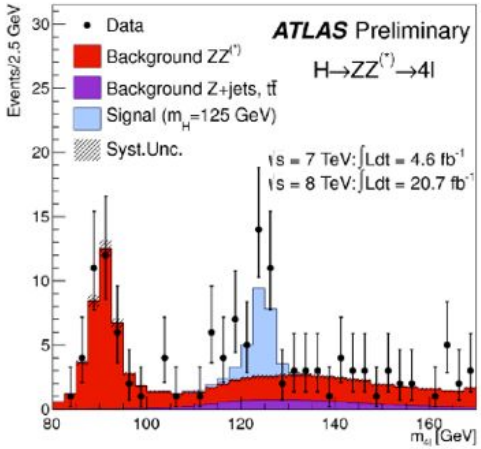
The detector uniformity is better than 2%.

Response of the calorimeter to single isolated tracks. To reduce the effect of noise, topological clusters are used in summing the energy.

T. LeCompte (ANL)

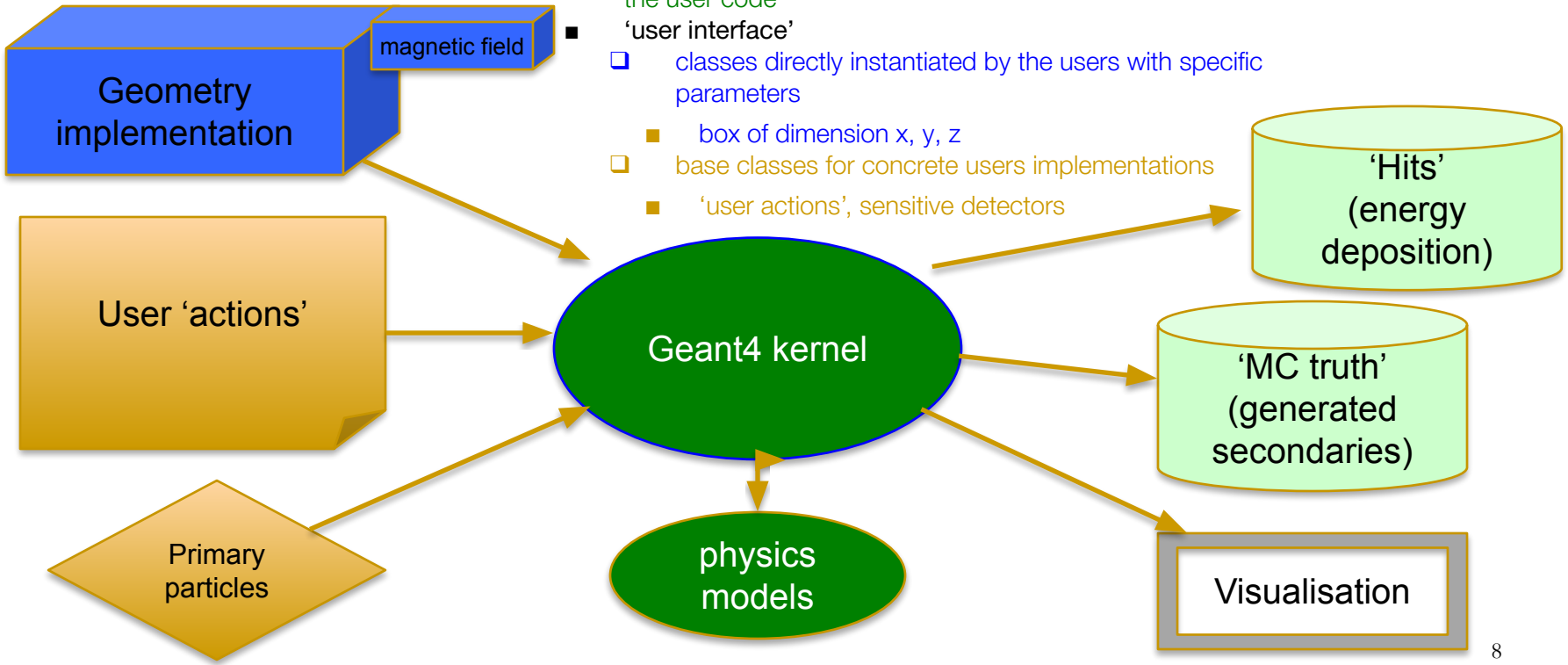
This plot agreed better than we ever expected. (I sent the student who made it back to make sure that they didn't accidentally compare G4 with G4.





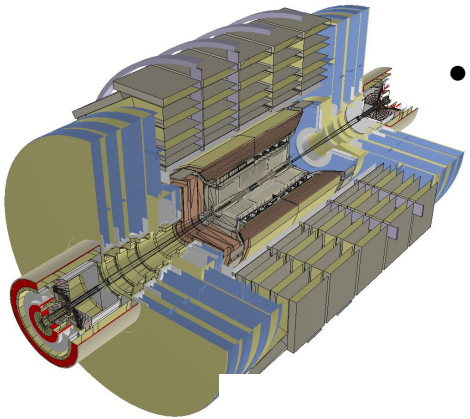
Geant4 application architecture

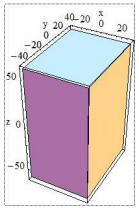
- 'kernel', internals of the engine, no direct interaction with the user code
- 'user interface'
  - classes directly instantiated by the users with specific parameters
    - box of dimension x, y, z
  - base classes for concrete users implementations
    - 'user actions', sensitive detectors

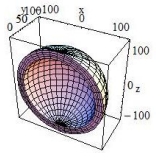


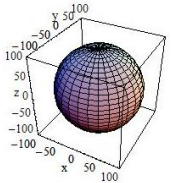
Geometry

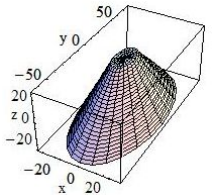
- Implemented using ‘lego bricks’ of different shapes
- Built as hierarchies of volumes

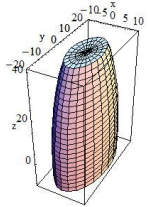


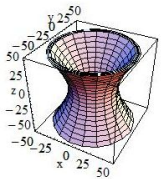


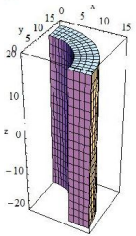


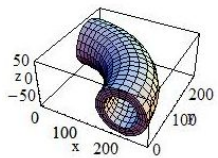


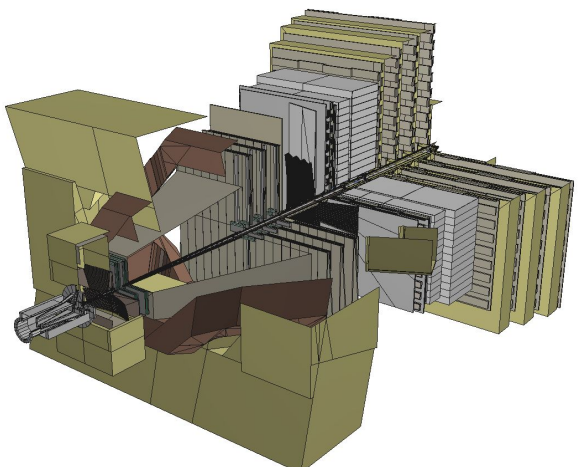


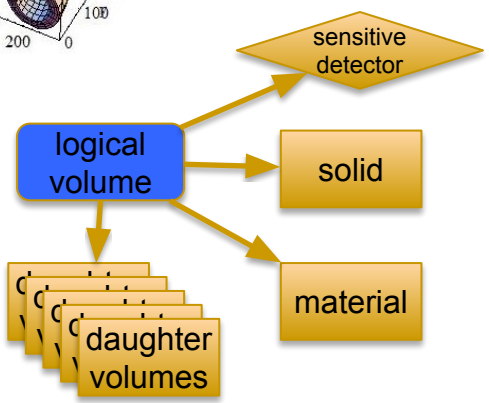


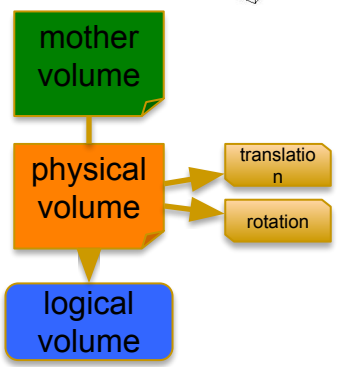






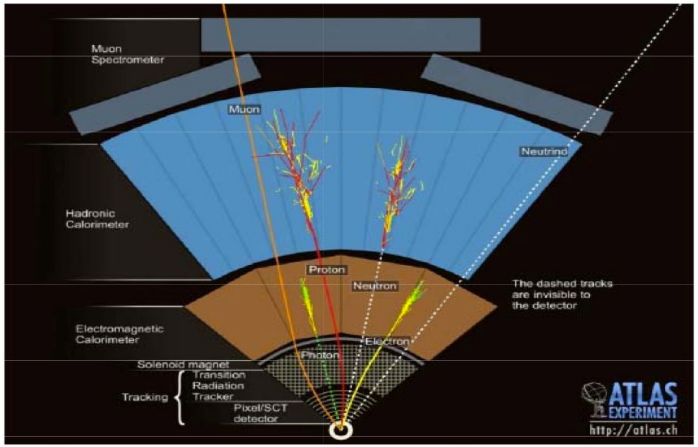


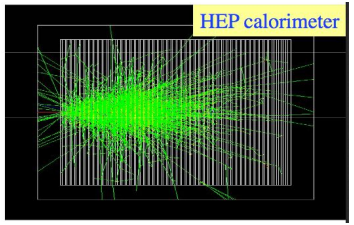




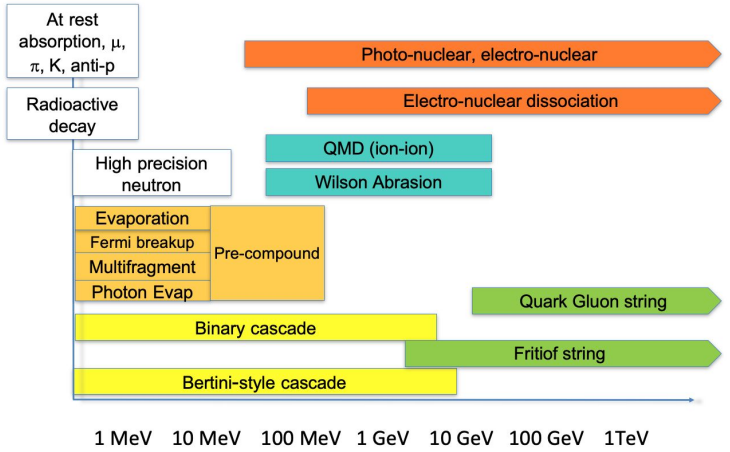
Geant4 physics

Geant4 implements the physics processes needed to simulate the response of the different subdetectors





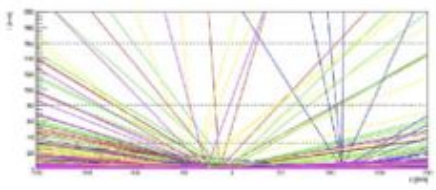
- Electromagnetic physics
  - Gammas:
    - Gamma-conversion, Compton scattering, Photo-electric effect
  - Leptons(e, μ), charged hadrons, ions
    - Energy loss (Ionisation, Bremstrahlung), Multiple scattering, Transition radiation, Synchrotron radiation, e+ annihilation.
  - Photons:
    - Cerenkov, Rayleigh, Reflection, Refraction, Absorption, Scintillation
- Hadronic physics

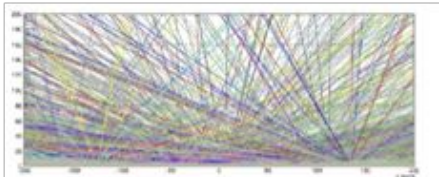


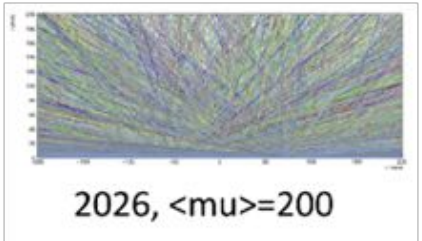
Needs of the experiments for future accelerators

- For 2021 data taking the LHC smaller experiments are moving to a 40 MHz data acquisition
  - Installation of upgrade detectors in progress
  - In LHCb full software trigger with high signal purity
- From 2026, the LHC will enter a new era with a 5-7x increase in the original design luminosity
  - Up to 200 pile up interactions per crossing in ATLAS and CMS
- New detectors with more channels
  - 4-5x increase in event size
- Upgraded trigger system
  - Up to 10x increase in the offline event rate

A 10+ year program of precision and discovery physics with a ten-fold increase in integrated luminosity

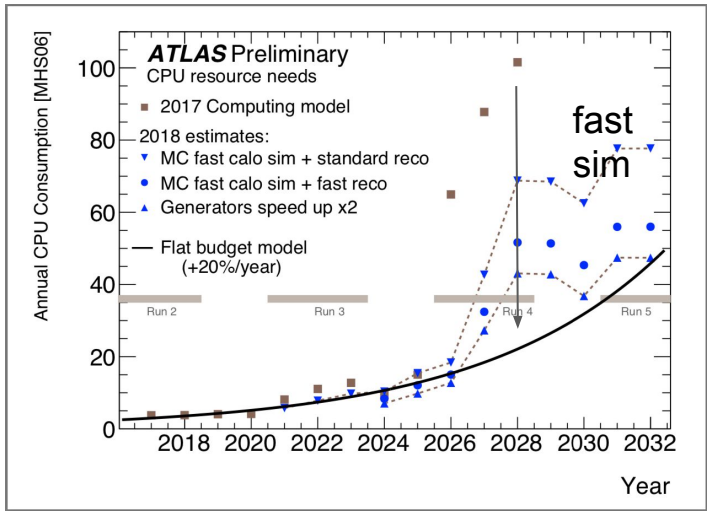


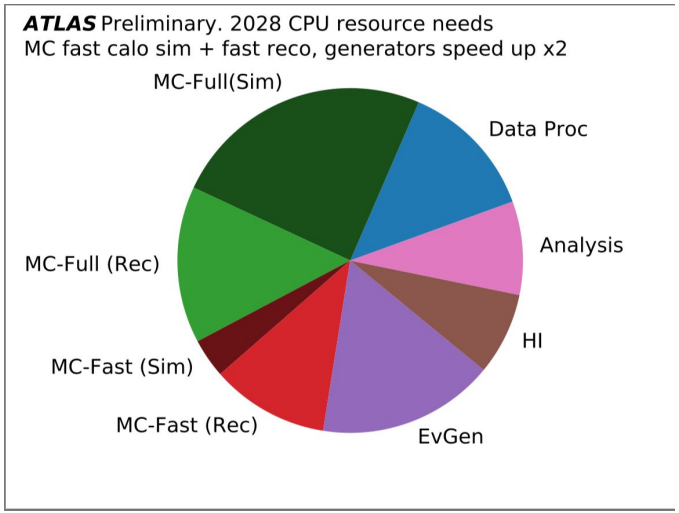




Needs of the experiments for future accelerators

- Flat-budget computing hardware improvements fall well short of requirements
  - CPU needs dominated by simulation
- Also, HEP software is too serial for future architectures
  - LHCb and ALICE are vectorizing their specific software and exploring the use of GPUs in trigger

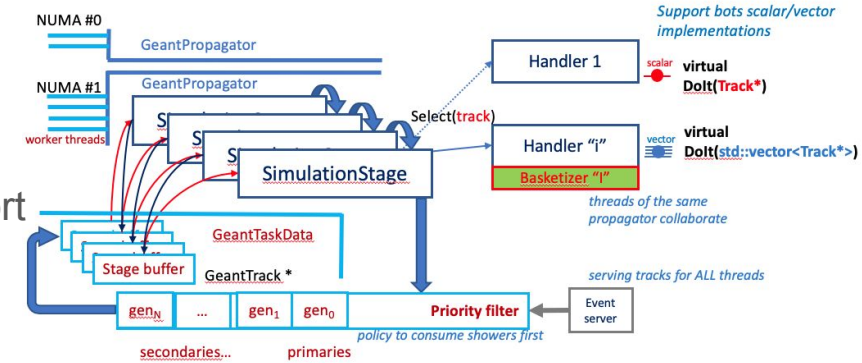




https://twiki.cern.ch/twiki/bin/view/AtlasPublic/ComputingandSoftwarePublicResults

GeantV R&D on vectorised transport

A generic vector flow approach

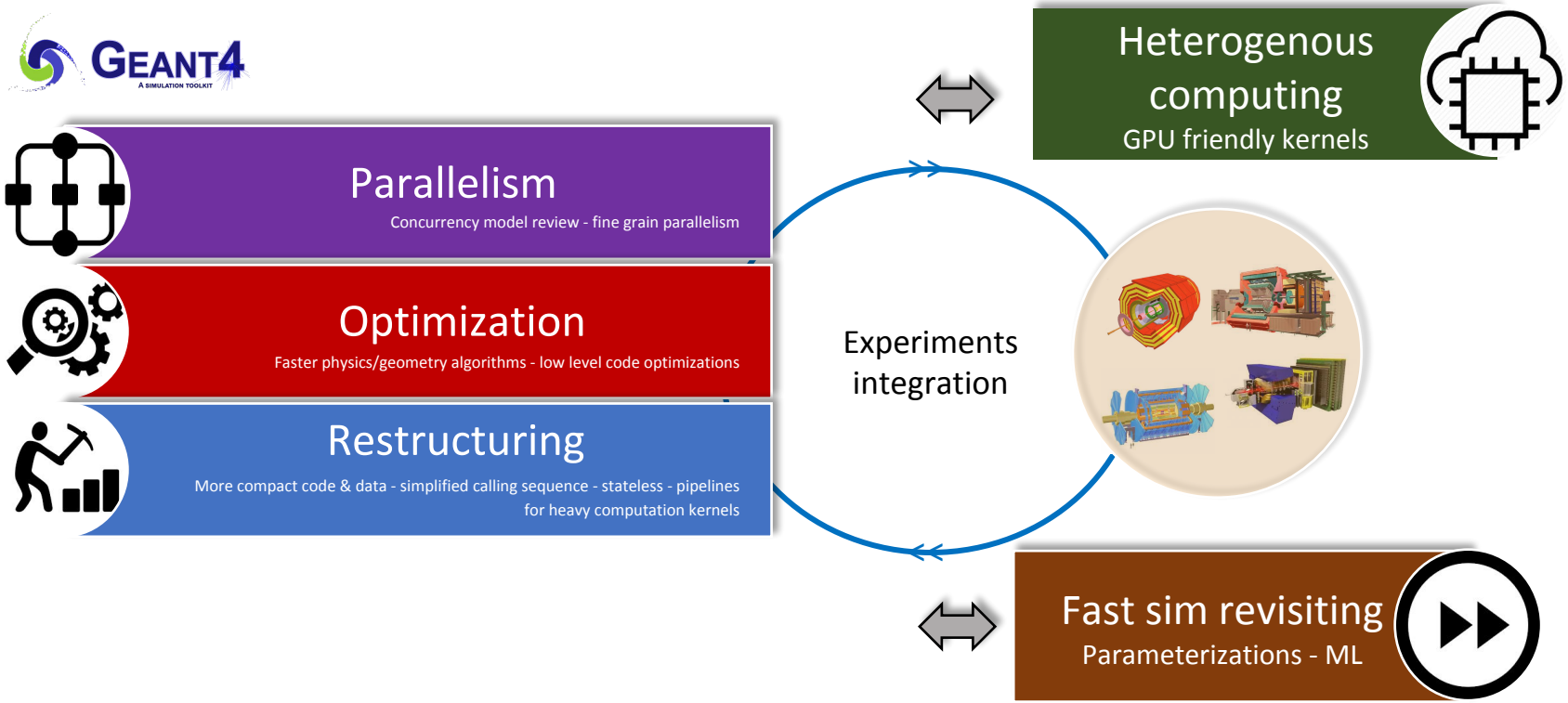


- GeantV R&D explored vectorized particle transport for next generation simulation toolkit
  - aimed at demonstrating the speed up of simulation using a novel approach to concurrent processing and data handling, exploiting vector operations on modern CPUs
- HSF community meeting held October 2019 with the outcome of the GeantV prototype
  - Prototype with EM physics was finalised. Many comparisons with equivalent Geant4 application done. Before the end of year publish a technical paper with all the details
- Libraries developed for the prototype are very useful (e.g. VecCore, VecGeom, VecMath)
  - Successfully integrated in Geant4, ROOT, etc.
- Vectorisation (organising the work in baskets of particles) does not bring the expected speedups. In some cases deteriorates the overall performance.
  - Large overheads in continuously reshuffling particles in baskets and dealing with the tails.
- Re-writing and modernising large parts of Geant4 potentially could bring us a factor of 2.0 ± 0.5 in performance (depending on the CPU/caches)
  - Compact code, better data formats, data locality, less virtual functions, etc.

Three Main Axes of Development

- **Improve, optimise and modernise** the existing Geant4 code to gain in performance for the detailed simulation
  - Re-structure the code to make possible major changes (task-oriented concurrency, specialisation of the physics, better data formats, etc.)
  - Some recent successes but we need to do much more
- Trade precision for performance using **fast simulation techniques** both with parameterisations and with ML methods, and integrate them seamlessly in Geant4
  - Use detailed simulation to ‘train networks’ or to ‘fit parameters’ that later can deliver approximative detector responses well integrated within Geant4
- Investigate the **use of ‘accelerators’** such as GPUs
  - With novel approaches for organising the computational work

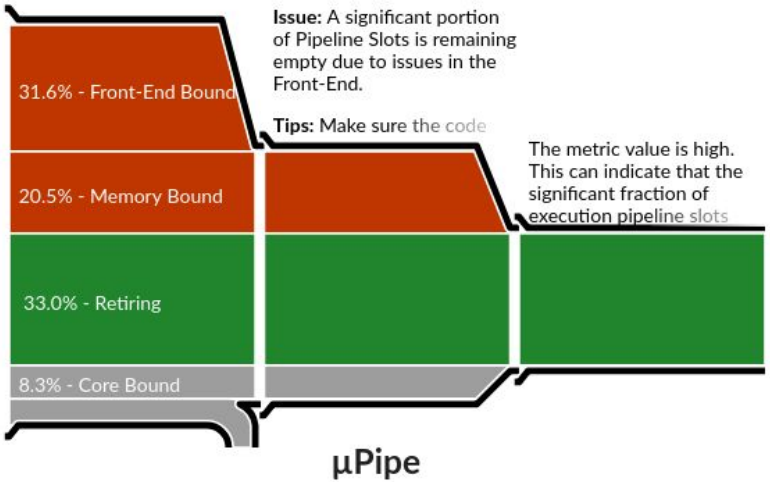
Performance: main directions



Ongoing Investigations

- Code compactness, simplified calling sequence, optimizations
  - A large part of the GeantV speed-up coming from better fitting the instruction cache
  - More streamlined computation for HEP simulation hotspots
- More flexible parallelism model, accelerator friendly
  - Sub-event parallelism, task parallelism
  - Efficient track-level parallelism on warps

Example of CPU µ-pipe for CMS EM shower simulation w/ Geant4

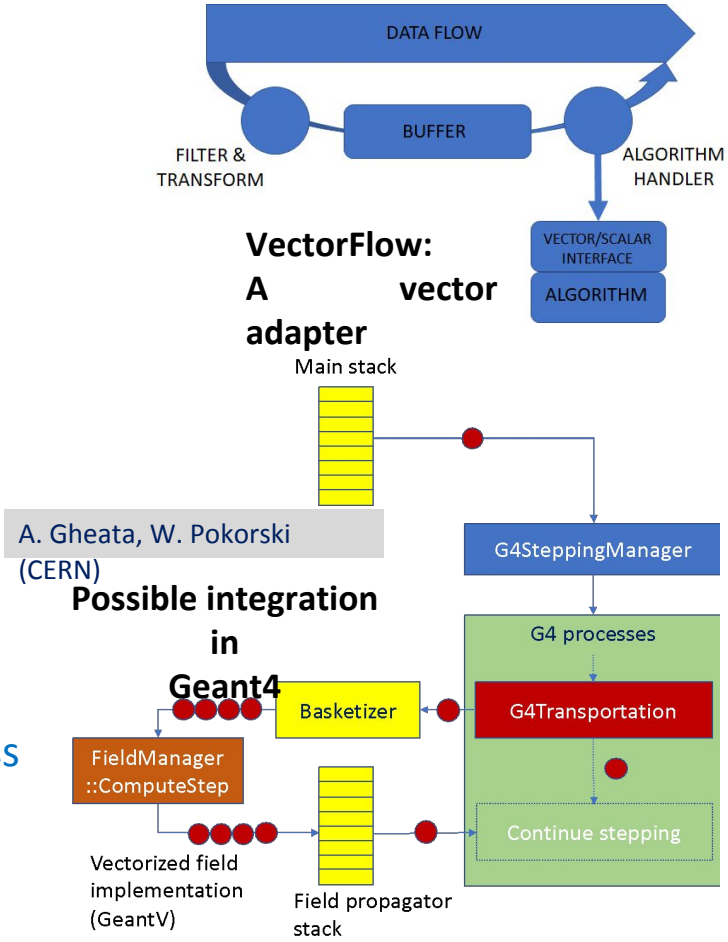


Vtune Microarchitecture analysis
Xeon®CPU E5-2630 v3@2.4 GHz

Vector pipelines in Geant4

Generalizing vectorization by passing vectors of data to functions rather than rely on inner loops.

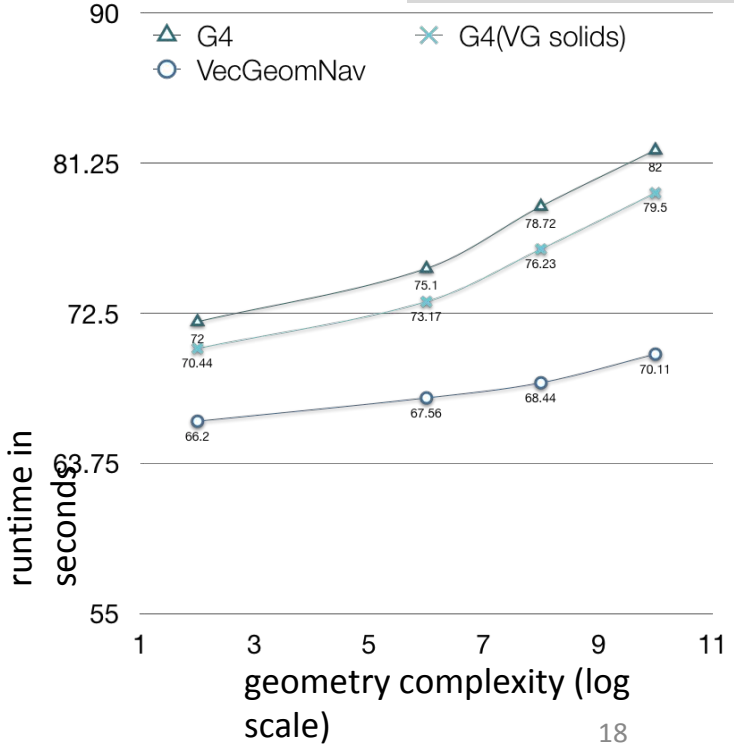
- Idea originating from GeantV workflow, but generalized as templated API usable in any workflow
  - Using VecCore as vectorization library
- Prototyping the changes needed in Geant4 for such extension
  - Ongoing work for making Geant4 transport stateless
  - Aiming to prototype integration w/ FP-intensive modules



Optimizing Geant4 navigation using VecGeom

- A first implementation of a Geant4 **navigation plugin**, using VecGeom capabilities.
  - Allows to make use of the modular and extensible navigation acceleration structures of VecGeom
- Tests on full detector geometries remain to be done and development to be completed
  - Preliminary tests on simplified geometry very promising: 10-15% speedup vs. G4 native geom
- Related ongoing work: VecGeom navigation specialization for some volume topologies

S. Wenzel (CERN)

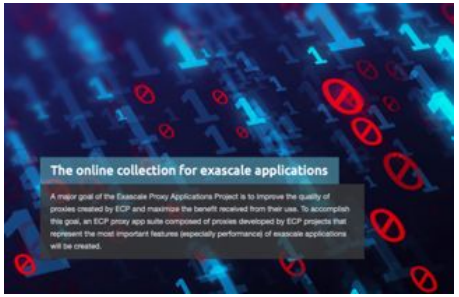


Task parallelism in Geant4

J. Madsen (LBL)

- Geant4 can benefit from having internally nested task-based parallelism
  - Making parallelism transparent to users (i.e no G4MTRunManager)
  - Better support for sub-event parallelism, eventually track-level parallelism
- Easier to expose simulation as a task
  - In relation with concurrent task based frameworks (e.g. CMSSW, Gaudi, …)
- First implementation of tasking support already available
  - gitlab.cern.ch/jmadsen/geant4-tasking
  - Based on standalone tasking library: github.com/jrmadsen/PTL
    - Native C++ features (future, promise, packaged_task, coroutines)
    - TBB backend available, PTL forwards task to TBB scheduler instead of internal
  - Support for multiple task pools
    - E.g for off-loading work to coprocessors

Exploring GPU usage in full HEP simulation

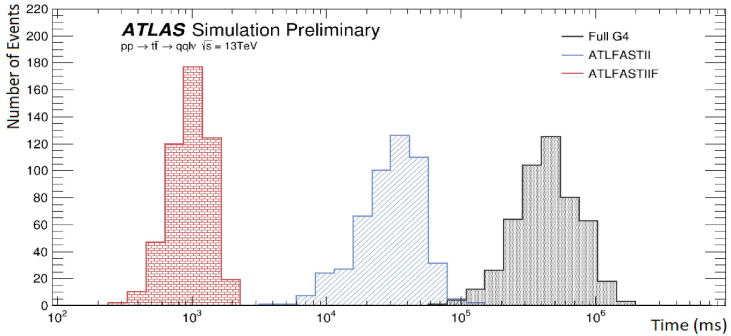


- Geant Exascale Pilot Project – several collaborators from US
  - Goal to **study and characterize architecture and performance** to best use GPUs in general and Exascale facility in particular for full HEP simulation
    - Explore memory access, computation ordering, and CPU/GPU communication patterns
    - Avoid over-simplification
    - Reuse or leverage existing packages, not bound by backward compatibility
  - Strategies
    - Focus on NVidia compiler at first (later look at Kokkos and others)
    - Research way to increase instruction and data cache efficiency
  - Early technical ideas
    - **Partial Static Polymorphism**: allow upload/download of data to device without transformation
    - **Separation Of State and Access and Functional Approach**: allow significant data memory layout change without code change

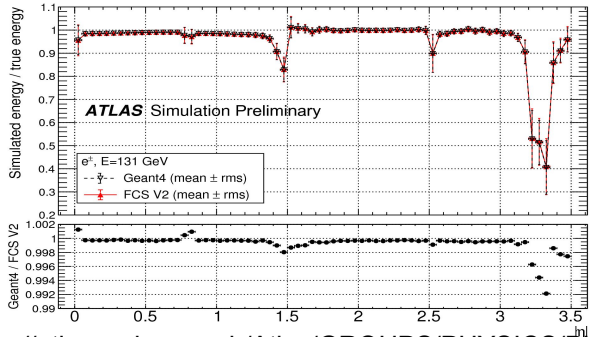
- GPU-aware physics code restructuring being investigated
  - Kernels for EM shower physics “confined” to GPU, w/o user code calls

Fast(er) Simulation

- Moving physics analyses from detailed to fast simulation is a critical assumption in computing models for [HL-]LHC, e.g. ATLAS as an example
  - FastCaloSim (parametrised calorimeter response) gains an order of magnitude over G4
  - FastChain (fast sim + fast reco) gains a further order of magnitude
- Fully parametric simulations to replace the whole simulation and reconstruction
- R&D into use of machine learning
- Also work is beginning towards deployment on HPCs and accelerators



https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PLOTS/SIM-2019-002/



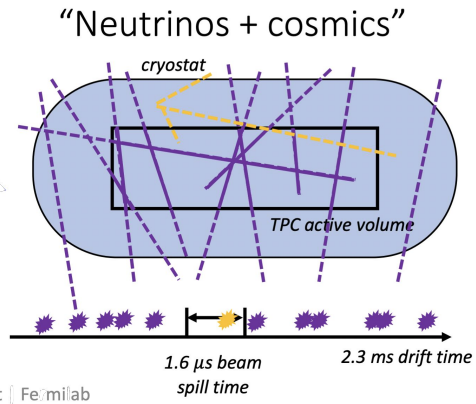
https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2018-002

Challenges in neutrino experiments

- The primary goal of many neutrino experiments is to measure neutrino oscillation parameters
- Interaction cross sections & detector uncertainties have significant impact
  - Determination of the incident neutrino is based upon interpretation through nuclear model of reconstructed final-state objects: tuning the models to the data is far from easy!
- Precision measurements requires accurate simulation of detector response and efficiency

- As an example the DUNE development of trigger, final detector design, shower reconstruction, and energy resolution depends upon photon simulations
- The magnitude of the problem is such that it is necessary to simulate 6M photons in the Liquid argon Far Detector on a CPU
- DUNE wants “natively GPU-accelerated optical photon tracking built into a fully-featured detector simulation”

https://indico.cern.ch/event/759388/contributions/3331550/



Conclusion

- Detector simulation plays a key role in the physics program of the LHC
- Primarily rely on the simulation toolkit: Geant4
  - Excellent precision
- However, as the dataset size is growing so are the simulation needs
  - LHC experiments are investing in a range of fast simulation techniques
  - Also, need ever greater precision
- Computing resources are evolving. Need to understand how the software can be adapted to fully exploit what will be available to the HEP community
  - Vectorization and parallelism
  - HPCs and accelerators, GPUs, …